WEST Search History

DATE: Friday, August 02, 2002

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L5	(usb near5 advantag$).bsum.	30	L5
L4	usb.bsum.	655	L4
L3	4754268[uref]	75	L3
L2	L1.ti.	10	L2
L1	wireless near5 (mouse or pointing device)	260	L1

END OF SEARCH HISTORY

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L6	L5.ti.	25	L6
L5	(pointing device or mouse) near5 integrat$	722	L5
L4	(pointing device or mouse) near5 hub near5 integrat$	0	L4
L3	5489922[uref]	14	L3
L2	L1.ti.	9	L2
L1	remot$ near5 (pointing device or mouse)	771	L1

END OF SEARCH HISTORY

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L17	l15 and (host)	0	L17
L16	l15 and (hub or usb)	0	L16
L15	6131130.pn.	1	L15
L14	l8 and (parallel or serial)	1	L14
L13	l8 and usb	0	L13
L12	l8 and hub	0	L12
L11	l1 and hub	31	L11
L10	5880721[uref]	6	L10
L9	5790201[uref]	21	L9
L8	5790201.pn.	1	L8
L7	(4837812| 5652766| 5668566| 5790201| 5854621| 5864300| 5880721)![pn]	7	L7
L6	L4 and wireless	17	L6
L5	L4 and wirless	0	L5
L4	elamin-$.xa.	136	L4
L3	5793359[uref]	36	L3
L2	L1.ti.	32	L2
L1	wireless near5 (keyboard or input device)	572	L1

END OF SEARCH HISTORY

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=ADJ		
L13	5371858.pn. and (synchron$ near10 asynchron$)	1	L13
L12	configu$ near3 port near5 (synchron$ near10 asynchron$)	8	L12
L11	L10.ab,ti.	0	L11
L10	configu$ near3 port near5 (synchron$ or asynchron$)	99	L10
L9	L6 near10 (synchron$ or asynchron$)	15	L9
L8	L6.ti.	2	L8
L7	L6 near5 (multiple or plurality)	11	L7
L6	serial port interface	1020	L6
L5	serial port interfacce	0	L5
L4	L1.bsum.	6	L4
L3	L1.ti,ab.	0	L3
L2	L1 near5 (multiple or plurality)	5	L2
L1	serial port adj (card or board)	32	L1

END OF SEARCH HISTORY

WEST

Generate Collection

L3: Entry 7 of 75 File: USPT

US-PAT-NO: 6304250
DOCUMENT-IDENTIFIER: US 6304250 B1

TITLE: Wire/wireless keyboard with pointing device attachable thereto, computer system for use with the same, and related method

DATE-ISSUED: October 16, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Yang; Chang-Hwan	Kyunggi-do			KR
Kim; Sung-Soo	Kyunggi-do			KR

US-CL-CURRENT: 345/168; 341/22, 341/23, 345/161, 345/163, 345/167, 345/169

WEST

Generate Collection Print

L5: Entry 12 of 48 File: USPT

DOCUMENT-IDENTIFIER: US 5999022 A
TITLE: Signal transmission driver circuit, receiver circuit, and method thereof for transmitting and receiving information based on multiple periods and/or a delay function

Detailed Description Text (88):
The signal transmission circuit according to the present invention has the advantage of high noise resistance and thus providing a noise margin. Moreover, since reference voltage potential and signal voltage potential (i.e., complementary date) can be transmitted by using only a single signal line, there is the advantage of requiring small layout area for the signal transmission circuit. Furthermore, since the amplitude of the signal transmitted through the single signal line is small, there is the advantage of small power consumption of the signal transmission circuit.

WEST

End of Result Set

☐ Generate Collection | Print

L13: Entry 1 of 1 File: USPT

DOCUMENT-IDENTIFIER: US 5371858 A
TITLE: Data communication system for assigning addresses to hand-held data terminals

US PATENT NO. (1):
5371858

Detailed Description Text (61):
The controller 400 may be configured as a communications controller. Four sync/async RS-323 ports are software configured for host and remote communication channels. A RS-323 port will enable host communications using the most common asynchronous and synchronous protocols. These include Bisync, two-way TTY and ADCCP. During system configuration, the target protocol is specific and soft-loaded from a removable disk inserted in the disk receptacle slot 224, FIG. 18, so as to be loaded in executable RAM.

Detailed Description Text (85):
Two 85C30 SCC's, SCC1 and SCC2, FIG. 30B, may be used for host and terminal communications. The four channels may be configured as synchronous or asynchronous, RS-323 ports. One of the four communication ports may be software configured for RS-485 LAN communications.

Detailed Description Text (106):
The Network Controller may provide both synchronous and asynchronous communications support for devices which are remote to the host site. This support may include auto-dial and auto-answer capabilities.

Detailed Description Text (127):
Several options are provided for support of remote sites. Network Controllers at remote sites have one RS232 "host" port, for communications with a "host" Network Controller, add one LAN port. The communications link to the host controller can be manually configured to use either ADCCP or a TTY extension as the data-link protocol. ADCCP ports on a host controller may communicate with either remote controllers configured for ADCCP or with ADCCP terminals in Multi-quad Lockboxes. TTY ports on a host controller may communicate with 4300/4400 terminals, 101/121/141 terminals or other TTY devices of Norand Corporation. The host controller determines if a remote device is a Network Controller or an ADCCP/TTY device, when the connection is made. The host may dynamically configure RS232 ports on a Network Controller as either ADCCP or TTY ports when the port is activated. This facilitates the use of a single port for synchronous communications to remote Network Controllers and asynchronous communications to TTY devices.

Detailed Description Text (206):
18. The host responds with an auto-answer activation record for the LAN port and port 2, and an auto-dial activation record for port 3. Port 2 is configured to be asynchronous and port 3 is configured to be synchronous. Connection requests are enabled for the LAN port.

WEST

Generate Collection | Print

also, small power consumption

by increasing simplicity and reliability of the system

DOCUMENT-IDENTIFIER: US 6219730 B1
TITLE: Method and apparatus for producing a combined data stream and recovering therefrom the respective user input stream and at least one additional input signal

Detailed Description Text (7):
In applications where the efficiency of the communication link is desired, such as in wireless communication, combined data stream 37 is encoded into a single signal transmitted by a single wire or electromagnetic wave. FIG. 4A depicts one embodiment of the invention in which framer 34 encodes data stream 37 into a single signal. In reference to FIG. 4A, the codec TP3054, in place of converter 30, converts output information 54, sent from the computer via line TXD, into output signal 52 entering output 53. Concurrently, the codec also converts input signal 31, delivered by input 32, into serial stream 33. In order to make stream 33 compatible with the RS-232's data format, a start-bit pulse provided by signal STA* is logically AND-ed with stream 33 to become digital input stream SX carrying information of the input signal. Since codec 30 converts output information 54 and input signal 31 simultaneously, their corresponding data streams traveling on the RS-232's data lines, TXD and RXD, are preferably synchronized for their timely transmission and reception by the computer. One logic implementation of the framer, as depicted in FIG. 4A, relies on the signal of line TXD to maintain the synchronization between the UI and input streams. The data lines TXD and RXD are programmed by the computer to have a data rate of 115.2 kbits/s and a frame format composed of eleven bits: one start-bit, eight data-bits, one parity-error-bit (PE-bit) and one stop-bit. In reference to FIGS. 4A and 4B, flip-flop FF2, utilizes signal CLR to re-synchronize clock generator 62 with the start-bit of signal TXD*. Signal CLR is kept low by signal R* during the transmission of each data frame of signal TXD*; otherwise, due to the high voltage VCC at FF2's input pin, it waits for being set high by the upcoming start-bit of signal TXD*. Once set, signal CLR restarts, and is immediately reset by the same, clock generator 62 providing bit-clock signal BCLK of the codec. Signal BCLK is divided by sixteen to become signal QH, which is latched into flip-flop FFI to provide the codec's frame-sync signal FS converting input signal 31 into serial stream 33 coming out of the codec's pin DX. Simultaneously, signal FS moves output information 54 serially into the codec, via pin DR, to be converted into output signal 52. Both UI stream 24 and input stream SX are synchronezed and encoded into a single signal RXD* by data selector 50 under control of signal SEL. The latter, formed by signals coming from clock generator 62, produces a string of positive pulses in sychrony with the PE-bit locations of signal RXD*, referred to as the UI-bit and denoted as "UI" in FIG. 4B. Data selector 50, normally selecting the input stream SX, otherwise samples UI stream 24 and stores its data into the UI-bit location of signal RXD*. Since UI stream 24 varies slowly compared to the pulsing of signal SEL, its information is conserved in, and subsequently can be recovered from, the UI-bits of signal RXD*. The UI-bit location on signal RXD* is not necessarily at the PE-bit. For instance, the least-significant data-bit (DO) location may be used instead. In such case, the pulses of signal SEL occur at the DO-bit time slots; consequently, the PE-bits, no longer needed, may be omitted entirely from signal RXD* to enhance the data transfer rate. In order to provide clocking signals, oscillator OSC of frequency around 1.8 MHz generates master clock MCLK running the codec. Clock generator 62 divides signal MCLK by sixteen to produce bit clock BCLK. The timing waveform for signals SEL, CLR, FS, QH, R*, TXD*, and RXD* are depicted in FIG. 4B, in relation to bit clock BCLK.

WEST

Generate Collection Print

bandwidth Saving along a particular Transmission path.

L4: Entry 9 of 12 File: USPT

DOCUMENT-IDENTIFIER: US 6377782 B1
TITLE: Method and apparatus for communicating between a client device and a linear broadband network

Detailed Description Text (6):
In a second embodiment of a subscriber access interface device 10 according to the teachings of the present invention, a plurality of the client devices 1 are connected to the subscriber access interface device 10 via a wireless connection. The client devices 1 modulate the upstream baseband signal 3 onto remote upstream wireless carriers to produce remote upstream modulated carriers 83. It is preferred that the remote connection between the client devices 1 and the subscriber access interface device 10 in a wireless communication embodiment follow the Home RF or Bluetooth communication protocols, but the IEEE 802.11 protocol is also an option. The remote upstream modulated carrier 83 is demodulated to reproduce the upstream data packets 24.

Detailed Description Text (12):
The connection from the initiating client device 1 to the subscriber access interface device 10 may be a wired connection or a wireless connection. In the case of a wireless connection between the initiating client device 1 and the subscriber access interface device 10, known communication protocols such as Home RF, Bluetooth, or IEEE 802.11 are appropriate. Specifications of the Home RF and Bluetooth communication protocols are hereby incorporated by reference. Accordingly, prior to further processing, the upstream information signal 11 is converted into the upstream baseband signal 3 having the IP format. The upstream baseband signal 3 comprises a series of upstream data packets 24. Each upstream data packet 24 comprises a service data unit ("SDU") 48 containing upstream information to be sent to the destination client device 71 with an address header 25 in a Transmission Control Protocol/User Data Protocol ("TCP/UDP") format. The address header 25 contains one or more destination addresses, the type of data in the SDU 48 (i.e. voice, video, or data), and the total number of bytes that make up the SDU 48. As one of ordinary skill in the art appreciates, the address header 25 contains the information necessary to route the upstream data packet 24 to the appropriate destination client device 71 in the same way connectivity is established on the Internet.

Detailed Description Text (25):
With further reference to FIG. 8 of the drawings, a primary upstream function of the subscriber access interface unit is gathering, encoding, and multiplexing the signals from a plurality of initiating client devices 832 (1 in FIG. 2 of the drawings) onto a single signal for wireless transmission to the NAID (6 in FIG. 1 of the drawings). FIG. 8 of the drawings represents a DSSS embodiment of the SAID in which a plurality of initiating client devices 832 generate upstream signals. Each upstream signal is encoded into an upstream baseband signal (3 in FIG. 3 of the drawings). The system CPU 826 receives each upstream baseband signal 3, prioritizes, multiplexes, and generates and appends the address header (25 in FIG. 4 of the drawings) and the multiplexer header (84 in FIG. 4 of the drawings) onto each packet before launching onto the system data bus 822 for storage into the system memory 828. The MAC protocol controller 824 receives the plurality of packets stored in the system memory 828 and encapsulates and then fragments the packet for transmission over the IEEE 802.11 wireless link. The digital baseband processor 820 clocks and

scrambles the downstream transmission fragments (89 in FIG. 4 of the drawings) and differentially encodes the downstream transmission fragments before applying a spread spectrum modulation. The digital baseband processor 820 quadrature phase shift key modulates each fragment to generate a baseband signal having I and Q components. The digital baseband processor 820 then spreads the I and Q symbols with a pseudo-random number sequence generator and sends it to a digital to analog converter 834 where the baseband signal is converted to an analog waveform. The analog waveform is transmitted to a quadrature IF modulator 836 that provides shaping and filtering and up-converts it into an IF signal. The gain controlled IF signal is further up-converted to the 2.4 to 2.5 GHz band by a transmission mixer 838. The output signal from the transmission mixer 838 is filtered at 840 and power controlled at 842 to an optimized output level before it is fed into the transmit/receive diversity switch 806. An optimum power level is application dependent and differs for each subscriber access interface unit in a system. The diversity switch 806 connects the output signal to a transmit antenna 844 for wireless transmission by the NAID and subsequent reception by the SAID. AMAC switch control line 846 controls the position of the diversity switch 806 to connect the transmit antenna 844 for delivery of the request to send signal. The MAC switch control line 846 then switches the position of the diversity switch 806 to connect the receive antenna 804 for reception of the clear to send signal. As described and shown in FIG. 8, the SAID 6 may comprise separate transmit and receive antennas 844, 804 respectively with a double pole/double throw diversity switch 806. Alternatively, the SAID may comprise a single transmit/receive antenna 848 for use with a single pole/double throw diversity switch 850. The MAC switch control line 846 controls the diversity switch 806, 850 similarly in both alternatives.

WEST

Generate Collection

L2: Entry 5 of 11 File: USPT

DOCUMENT-IDENTIFIER: US 5805931 A
TITLE: Programmable bandwidth I/O port and a communication interface using the same port having a plurality of serial access memories capable of being configured for a variety of protocols

CLAIMS:

1. A communications interface having a plurality of I/O ports each of which has a programmable bandwidth for allowing asynchronous data transfer through said plurality of ports, said communications interface comprising:

2. A communications interface having a plurality of I/O ports each of which has a programmable bandwidth for allowing asynchronous data transfer through said plurality of ports, said communications interface comprising:

WEST

Generate Collection Print

L2: Entry 2 of 8 File: USPT

DOCUMENT-IDENTIFIER: US 6320855 B1
TITLE: Method and system for initiating idle handoff in a wireless communications system

Brief Summary Text (9):
In a wireless communications channel, the presence of obstacles in the environment such as buildings, trees, mountains, cars and the like often results in the reflection of wireless communications signals transmitted by either mobile or base stations. This phenomenon is referred to as a multipath propagation environment because any particular wireless communications receiver (such as a mobile station) may receive a plurality of signals corresponding to a single signal transmitted by a particular wireless communications transmitter (such as one or more base stations), each of the plurality of received signals having traveled a different path to the receiver. Typically, the mobile radio channel also is a time varying multipath channel. In other words, the stream of pulses that would be received following the transmission of an ideal pulse over a mobile radio channel would change in time location, attenuation and phase depending on when the ideal pulse were transmitted. This is due in part to relative motion between the wireless transmitters and the environmental obstacles. It also due in part to fading, which occurs when multipath signals are phase shifted to such a degree that destructive interference with one another occurs, and path loss, which is a result of atmospheric effects on wireless communications signals.

claim 28/28

WEST

Generate Collection Print

Claim 2

L1: Entry 4 of 28 File: USPT

DOCUMENT-IDENTIFIER: US 6389029 B1
TITLE: Local area network incorporating universal serial bus protocol

Brief Summary Text (15):
Information is carried on the Universal Serial Bus in packets ("USB packets"). Packets sent at the low speed are called low speed transmissions. Similarly, packets sent at the full speed are called full speed transmissions. Each USB packet transmitted on the Universal Serial Bus is delineated by sync fields (for clock recovery) at the start of each USB packet, followed by the USB packet, and ending with a special end of USB packet signalling on the Bus. Referring to FIGS. 2A, 2B, 2C, 2D and 2E, the USB protocol supports five different main types of USB packets: a token packet, a start of frame packet, a data packet, a handshake packet and a special low speed preamble packet. At the beginning of each USB packet is a packet identifier or PID. According to the USB protocol, there are ten different types of PID's.

Brief Summary Text (16):
USB packets are sent within a plurality of transmission frames. Each frame is one millisecond long. Referring to FIG. 2B, start of frame packets are issued from the USB host software according to a precise one millisecond schedule. Each start of frame packet consists of a start of frame PID, a frame number and a CRC for error checking.

Brief Summary Text (17):
Data is carried on the Universal Serial Bus through the use of USB transactions. A USB transaction involves transmission of up to three USB packets for full speed transmissions and four packets for low speed transmissions. The USB host software formats the data destined to the USB devices into USB packets according to the USB protocol. (Described in more detail below). Similarly, each USB device formats data destined to the host computer into USB packets according to the USB protocol. (Described in more detail below).

Brief Summary Text (48):
In accordance with one aspect of the present invention, there is provided a computer network comprising a LAN hub, at least one network device connected to the LAN hub, at least one outer hub device connected to the LAN hub via a respective LAN link and at least one USB device or at least one LAN computer connected to the outer hub device via a respective USB link. The USB device or the LAN computer communicates with the outer hub device using the USB protocol. The outer hub device communicates with the LAN hub using the LAN protocol. The network device communicates with the LAN hub using the LAN protocol or a network protocol. For asynchronous communications, the outer hub device examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. The outer hub device buffers data received from the LAN hub to be sent to the USB device in a data packet and ensures that the data packet follows an Out token packet within the USB time limit prescribed by the USB protocol. Furthermore, the outer hub device buffers data received from the LAN hub to be sent to the LAN computer in a data packet and ensures that the data packet is sent to the LAN computer within the USB time limit prescribed by the USB protocol after receiving an In token packet from

the LAN computer.

Brief Summary Text (49):
In accordance with another aspect of the present invention, there is provided a computer network comprising a LAN hub, at least one outer hub device connected to the LAN hub via a respective LAN link, at least one other outer hub device connected to the LAN hub via a respective LAN link, at least one USB device or at least one LAN computer connected to the outer hub device via a respective USB link and at least one other LAN computer connected to the other outer hub device via a respective USB link. The USB device and the LAN computer communicates with the outer hub device using the USB protocol. The other LAN computer communicates with the other outer hub device using the USB protocol. The outer hub device and the other outer hub device communicates with the LAN hub using a LAN protocol. For asynchronous communications, the outer hub device examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the outer hub device buffers data received from the LAN hub to be sent to the USB device in a data packet and ensures that the data packet follows an Out token packet within the USB time limit prescribed time limit prescribed by the USB protocol. Furthermore, the outer hub device buffers data received from the LAN hub to be sent to the LAN computer in a data packet and ensures that the data packet is sent to the LAN computer within the USB time limit prescribed by the USB protocol after receiving an In token packet from the LAN computer. For asynchronous communications, the other outer hub device examines USB packets from the other LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within the USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the other outer hub device buffers data received from the LAN hub to be sent to the other LAN computer in a data packet and ensures that the data packet is sent to the other LAN computer within the USB time limit prescribed by the USB protocol after receiving an In token packet from the other LAN computer.

Brief Summary Text (50):
In accordance with another aspect of the present invention, there is provided an end hub for use in a computer network in which the end hub communicates with at least one USB device using the USB protocol and in which the end hub communicates with a LAN hub using a LAN protocol. The end hub comprises LAN hub communication means for communicating with the LAN hub, USB device communication means for communicating with the USB device and control logic means connected to the LAN hub communication means and to the USB device communication means. For asynchronous communications, the control logic means examines USB packets from the USB device, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition the control logic means buffers data received from the LAN hub to be sent to the USB device in a data packet and ensures that the data packet follows an Out token packet within the USB time limit prescribed by the USB protocol.

Brief Summary Text (51):
In accordance with the another aspect of the present invention, there is provided an attachment unit for use in a computer network in which the attachment unit communicates with at least one LAN computer using the USB protocol and in which the attachment unit communicates with a LAN hub using a LAN protocol. The attachment unit comprises LAN hub communication means for communicating with the LAN hub, USB computer communication means for communicating with the LAN computer and control logic means connected to the LAN hub communication means and to the USB computer communication means. For asynchronous communications, the control logic means examines USB packets from the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the control logic means buffers data received from the LAN hub to be sent to the LAN computer in a data packet and ensures that the data packet is sent to the LAN computer within the USB time limit

prescribed by the USB protocol after receiving an In token packet from the LAN computer.

Brief Summary Text (52):
In accordance with another aspect of the present invention, there is provided an enhanced attachment unit for use in a computer network in which the enhanced attachment unit communicates with at least one LAN computer using the USB protocol and in which the attachment unit communicates with a LAN hub using a LAN protocol. The attachment unit comprises LAN hub communication means for communicating with the LAN hub, USB computer communication means for communicating with the LAN computer and control logic means connected to the LAN hub communication means and to the USB computer communication means. The control logic means contains logic to virtually connect at least one USB device. For asynchronous communications, the control logic means examines USB packets from the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the control logic means buffers data received from the LAN hub to be sent to the LAN computer in a data packet and ensures that the data packet is sent to the LAN computer within the USB time limit prescribed by the USB protocol after receiving an In token packet from the LAN computer.

Brief Summary Text (53):
In accordance with another aspect of the present invention, there is provided a composite end hub for use in a computer network in which the composite end hub communicates with a LAN computer and with at least one USB device using USB protocol and in which the attachment unit communicates with a LAN hub using a LAN protocol. The composite end hub comprises LAN hub communication means for communicating with the LAN hub, USB device communication means for communicating with the at least one USB device, USB computer communication means for communicating with the LAN computer and control logic means connected to the LAN hub communication means, to the USB device communication means and to the USB computer communication means. For asynchronous communications, the control logic means examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the control logic means buffers data received from the LAN hub to be sent to the USB device in a data packet and ensures that the data packet follows an Out token packet within the USB time limit prescribed by the USB protocol. Furthermore, the control logic means buffers data received from the LAN hub to be sent to the LAN computer in a data packet and ensures that the data packet is sent to the LAN computer within the USB time limit prescribed by the USB protocol after receiving an In token packet.

Brief Summary Text (54):
In accordance with another aspect of the present invention, there is provided a virtual modem bridge for use with a first USB host device and a second USB host device in which the virtual modem bridge communicates with the first USB host device and the second USB host device using the USB protocol. The virtual modem bridge comprises first USB host device communication means for communicating with the first USB host device, second USB host device communication means for communicating with the second USB host device and control logic means connected to the first USB host device communication means and to the second USB host device communication means. For asynchronous communications, the control logic means examines USB packets from the first USB host device or the second USB host device, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets. In addition, the control logic means buffers data received from the first USB host device to be sent to the second USB host device in a data packet and ensures that the data packet is sent to the second USB host device within the USB time limit prescribed by the USB protocol after receiving an In token packet from the second USB host device. Furthermore, the control logic means buffers data received from the second USB host device to be sent to the first USB host device in a data packet and ensures that the data packet is sent to the first USB host device within the USB time limit prescribed by the USB

protocol after receiving an In token packet from the first USB host device.

Brief Summary Text (55):
In accordance with another aspect of the present invention, there is provided a method of increasing the distance between a host computer and a USB device where the host computer communicates with the USB device using the USB protocol, said method for sending data from the host computer to the USB device comprises transmitting an Out token packet and a data packet (collectively the "USB packets") from the host computer to a first outer hub device using the USB protocol; for asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the host computer within a USB time limit prescribed by the USB protocol after receiving the USB packets; for asynchronous communications, re-transmitting the USB packets from the host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the host computer or if no handshake packet was received by the host computer within the USB time limit prescribed by the USB protocol; converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device; transmitting the LAN packet from the first outer hub device to a second outer hub device using the LAN protocol; reconstructing the USB packets from the LAN packet at the second outer hub device; transmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the token packet within the USB time limit prescribed by the USB protocol; for asynchronous communications, examining the USB packets at the USB device, generating an appropriate handshake packet at the USB device according to the USB protocol and transmitting the handshake packet from the USB device to the second outer hub device within the USB time limit prescribed by the USB protocol after receiving the USB packets; for asynchronous communications, re-transmitting the USB packets using the USB protocol from the second outer hub device to the USB device ensuring that the data packet follows the token packet within the USB time limit prescribed by the USB protocol if a NAK handshake packet is received by the second outer hub device or if no handshake packet was received by the second outer hub device within the USB time limit prescribed by the USB protocol. In addition, said method for obtaining data from the USB device for the host computer comprises transmitting an In token packet from the host computer to a first outer hub device using the USB protocol; if the first outer hub device has data responsive to the In token packet performing the following steps: transmitting a USB data packet from the first outer hub device to the host computer using the USB protocol, and for asynchronous communications, examining the USB data packet at the host according to the USB protocol, generating an appropriate handshake packet at the host computer according to the USB protocol and transmitting the handshake packet from the host computer to the first outer hub device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB data packet; if the first outer hub device does not have data response to the In token packet, transmitting a NAK handshake packet from the first outer hub device to the host computer; if the first outer hub device has received a LAN packet indicating a stall, transmitting a stall handshake packet from the first outer hub device to the host computer; converting the In token packet into at least one LAN packet according to the LAN protocol at the first outer hub device; transmitting the LAN packet from the first outer hub device to a second outer hub device using the LAN protocol; reconstructing the In token packet from the LAN packet at the second outer hub device; transmitting the In token packet from the second outer hub device to the USB device using the USB protocol; if the USB device has data responsive to the In token packet, performing the following steps: transmitting the data in a data packet from the USB device to the second outer hub device using the USB protocol, for asynchronous communications, examining the data packet at the second outer hub device, generating an appropriate handshake packet at the second outer hub device according to the USB protocol and transmitting the handshake packet from the second outer hub device to the USB device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB data packet, and converting the data packet to at least one LAN packet according to the LAN protocol at the second outer hub device; if the USB device does not have data responsive to the IN token packet, transmitting a NAK handshake packet from the USB device to the second outer hub device using the USB protocol and converting the NAK handshake packet into at least one LAN packet according to the LAN protocol; if the USB device is in a

stalled condition, transmitting a Stall handshake packet from the USB device to the second outer hub device using the USB protocol and converting the Stall handshake packet into at least one LAN packet according to the LAN protocol; transmitting the LAN packet from the second outer hub device to the first outer hub device using the LAN protocol; reconstructing the data packet, the NAK handshake packet or the Stall handshake packet from the LAN packet at the first outer hub device; and, storing the data packet, the NAK handshake packet or the stall handshake packet in the first outer hub device.

Brief Summary Text (56):
In accordance with another aspect of the present invention, there is provided a method of increasing the distances in a computer network between a host computer and a USB device where the host computer communicates with the USB device using USB protocol, said method for sending data from the host computer to the USB device comprises transmitting an Out token packet and a data packet (collectively the "USB packets") from the host computer to a first outer hub device using the USB protocol; for asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the host computer using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB packets; for asynchronous communications, re-transmitting the USB packets from the host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the host computer or if no handshake packet was received by the host computer within the USB time limit prescribed by the USB protocol; converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device; transmitting the LAN packet from the first outer hub device to a LAN hub using the LAN protocol; re-transmitting the LAN packet from the LAN hub to a second outer hub device using the LAN protocol; reconstructing the USB packets from the LAN packet at the second outer hub device; transmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the Out token packet within the USB time limit prescribed by the USB protocol; for asynchronous communications, examining the USB packets at the USB device, generating an appropriate handshake packet at the USB device according to the USB protocol and transmitting the handshake packet from the USB device to the second outer hub device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB packets; and, for asynchronous communications, re-transmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the token packet within the USB time limit prescribed by the USB protocol if a NAK handshake packet is received by the second outer hub device or if no handshake packet was received by the second outer hub device within the USB time limit prescribed by the USB protocol. And said method for obtaining data from the USB device for the host computer comprises transmitting an In token packet from the host computer to the first outer hub device using the USB protocol; if the first outer hub device has data responsive to the In token packet performing the following steps: transmitting a data packet from the first outer hub device to the host computer using the USB protocol, and, for asynchronous communications, examining the USB data packet at the host computer, generating an appropriate handshake packet at the host computer according to the USB protocol and transmitting the handshake packet from the host computer to the first outer hub device using the USB protocol within a USB time limit prescribed by the USB protocol after receiving the USB data packet; converting the In token packet into at least one LAN packet according to the LAN protocol at the first outer hub device; transmitting the LAN packet from the first outer hub device to the LAN hub using the LAN protocol; re-transmitting the LAN packet from the LAN hub to the second outer hub device using the LAN protocol; reconstructing the In token packet from the LAN packet at the second outer hub device; transmitting the In token packet from the second outer hub device to the USB device using the USB packet; if the USB device has data responsive to the In token packet, performing the following steps: transmitting the data in a data packet from the USB device to the second outer hub device using the USB protocol, for asynchronous communications, examining the data packet at the second outer hub device, generating an appropriate handshake packet at the second outer hub device according to the USB protocol and transmitting the handshake packet from the second outer hub device to the USB device using the USB protocol within the USB time limit

prescribed by the USB protocol after receiving the data USB packet, and, converting the data packet to at least one LAN packet at the second outer hub device; if the USB device does not have data responsive to the IN token packet, transmitting a NAK handshake packet from the USB device to the second outer hub device using the USB protocol and converting the NAK handshake packet into at least one LAN packet according to the LAN protocol; if the USB device is in a stalled condition, transmitting a Stall handshake packet from the USB device to the second outer hub device using the USB protocol and converting the Stall handshake packet into at least one LAN packet according to the LAN protocol; transmitting the LAN packet from the second outer hub device to the LAN hub using the LAN protocol; re-transmitting the LAN packet from the LAN hub to the first outer hub device using the LAN protocol; reconstructing the data packet, the NAK handshake packet or the Stall handshake packet from the LAN packet at the first outer hub device; and, storing the data packet, the NAK handshake packet or the stall handshake packet in the first outer hub device.

Brief Summary Text (57):
In accordance with another aspect of the present invention, there is provided a method of increasing the distances in a computer network between a first host computer and a second host computer where the first host computer communicates with the second host computer using USB protocol, said method for sending data from the first host computer to the second host computer comprises transmitting a token packet and a data packet (collectively the "USB packets") from the first host computer to a first outer hub device using the USB protocol; for asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the first host computer using the USB protocol within a USB time limit prescribed by the USB protocol after receiving the USB packets; for asynchronous communications, re-transmitting the USB packets from the first host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the first host computer or if no handshake packet was received by the first host computer within the USB time limit prescribed by the USB protocol; converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device; transmitting the LAN packet from the first outer hub device to a LAN hub using the LAN protocol; re-transmitting the LAN packet from the LAN hub to a second outer hub device using the LAN protocol; reconstructing the USB packets from the LAN packet at the second outer hub device; in response to an In token packet sent from the second host computer to the second outer hub device using the USB protocol, transmitting the USB data packet from the second outer hub device to the second host computer using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the In token packet; and, for asynchronous communications, examining the USB data packet at the second host computer, generating an appropriate handshake packet at the second host computer according to the USB protocol and transmitting the handshake packet from the second host computer to the second outer hub device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB data packet.

Brief Summary Text (58):
In accordance with another aspect of the present invention, there is provided a method for allowing a first USB host device and a second USB host device to communicate with each other using USB protocol, said method for sending data from the first USB host device to the second USB host device comprises transmitting a token packet and a data packet (collectively the "USB packets") from the first USB host device to a virtual modem bridge using the USB protocol; for asynchronous communications, examining the USB packets at the virtual modem bridge, generating an appropriate handshake packet at the virtual modem bridge according to the USB protocol and transmitting the handshake packet from the virtual modem bridge to the first USB host device using the USB protocol within a USB time limit prescribed by the USB protocol after receiving the USB packets; for asynchronous communications, re-transmitting the USB packets from the first USB host computer to the virtual modem bridge using the USB protocol if a NAK handshake packet is received by the first USB host device or if no handshake packet was received by the first USB host device within the USB time limit prescribed by the USB protocol; in response to an In token packet sent from the second USB host device to the virtual modem bridge

using the USB protocol, transmitting the USB data packet from the virtual modem bridge to the second USB host device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the In token packet; and, for asynchronous communications, examining the USB data packet at the second USB host device, generating an appropriate handshake packet at the second USB host device according to the USB protocol and transmitting the handshake packet from the second USB host device to the virtual modem bridge using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB data packet.

Detailed Description Text (40):
The USB host software in each network device 40 translates data from client software into USB packets. Network protocol software in each network device 40 encapsulates the USB packets within the conventional network protocol according to the sub-protocol. Similarly, the network protocol software in each network device 40 extracts the USB packets from the network packets sent from the LAN hub 10 according to the sub-protocol. The USB host software extracts information or data from the USB packets. The information and the data is typically carried to the client software.

Detailed Description Text (47):
LAN packets from an outer hub device are received by the transceiver 320 of the LAN interface device 315 associated with the outer hub device and placed in the received buffer 470 of the bus-transceiver interface logic device 330. The microprocessor 310 moves the LAN packets in the received buffer 470 of the bus-transceiver interface logic device 330 via control unit 430 into the RAM 360. The microprocessor 310 converts the LAN packets into USB packets. The microprocessor 310 encapsulates the USB packets within network packets of the conventional network protocol according to the sub-protocol. The microprocessor 310 moves the network packets from the RAM 360 into the transmit buffer 480 of the network interface device 380 serving the network 20 via the control unit 450 of the network interface device 380.

Detailed Description Text (95):
Another aspect of this invention relates attaching host computers 130 to the LAN hub 10, not over established networks 20, but through the mediation of attachment units 110 as shown in FIG. 7. In this arrangement, the LAN hub 10 appears as a USB device connected to a USB port 150 of the host computer 130. The attachment device 110 is defined to appear to the host computer 140 as a special kind of modem (i.e. virtual modem) or network interface unit and normally operates at the full speed (12 Mbs). FIG. 12 shows how the attachment unit 110 appears to the connected host computer 130. The attachment unit 110 typically has 3 end points, end point 0, end point 1 and end point 2. As usual, end point 0 (sometimes called control end point 0) is used to configured the attachment unit 110. End point 1 is typically defined as the end point which receives data from the host computer 130. End point 2 is typically defined as the end point which sends data from the attachment unit 110 to the host computer 130. End points 1 and 2 typically carry bulk transactions. According to the USB specification 1.0, each bulk transaction can not exceed 64 bytes. The attachment unit 110 has an In data buffer (or a receive buffer) to receive data sent to the end point 1. The attachment unit 110 also has an Out data buffer (or a transmit buffer) which stores data to be sent from the endpoint 1 of the attachment unit 110. (discussed in more detail later) When the LAN computer 130 wishes to communicate with another LAN computer 130 or with a LAN computer 190, 215, 260 or a network device 40 (such as a remote computer), the client software in the LAN computer 130 typically generates IP packets according to the IP protocol. (Other packet protocols such as Ethernet can be used). Referring to FIG. 11F, since each IP packet is typically greater than 64 bytes, the USB host software fragments the IP packet 8000 into a plurality of USB packets 8010. The USB packets 8010 are sent to the attachment unit 110. The LAN hub sends LAN packets (encapsulating the USB packets 8010) to the LAN hub 10. If the IP packet 8000 is destined to a network device 40, the LAN hub 10 reassembles the IP packet and forwards it to the network device 40 in one or more network packets. If the IP packet 8000 is destined to a LAN computer 190, 215 or 290 or another LAN computer 130, the LAN hub 10 forwards the LAN packets (encapsulating the USB packets 8010) to the respective outer hub device servicing the respective LAN computer 190, 215, 290 or 130.

Detailed Description Text (96):
Similarly, referring to FIG. 11G, when the LAN computer 130 receives USB packets

8020 sent from another LAN computer 130, a LAN computer 215, 190, or 260 or a network device 40 (such as a remote computer), the USB host software reconstructs an IP packet 8030 from the USB packets 8020.

Detailed Description Text (97):
The LAN computer 130 can also communicate with a USB device 100 or 180 by addressing the LAN hub 10 in the IP (or Ethernet) protocol and encapsulating the USB protocol within the IP (or Ethernet) protocol. (i.e. A plurality of USB packets destined to the USB device 100 or 180 ("USB device packets") are sent in a plurality of IP (or Ethernet) packets). Similarly, referring to FIG. 11F, since each IP packet is typically greater than 64 bytes, the USB host software fragments up the IP packet 8000 into a plurality of USB packets 8010. The USB packets 8010 are sent to the attachment unit 110. The attachment unit 110 sends LAN packets (encapsulating the USB packets 8010) to the LAN hub 10. The LAN hub 10 reconstructs the IP packet 8000 from the plurality of USB packets 8010. The LAN hub also extracts the USB protocol from the IP (or Ethernet) protocol (i.e. The USB device packets are extracted from the IP (or Ethernet) packets). The LAN hub 10 creates and forwards LAN packets encapsulating the USB device packets to the end hub 80 serving the USB device 100 (or the composite end hub 160 serving the USB device 180).

Detailed Description Text (98):
Similarly, when the LAN hub 10 receives LAN packets encapsulating USB device packets from the end hub 80 (or the composite end hub 160), the LAN hub 10 extracts the USB device packets from the LAN packets and creates IP packets 8030 encapsulating the USB device packets. Since each IP packet 8030 is typically greater than 64 bytes, the LAN hub 10 fragments the IP packet 8010 into a plurality of USB packets 8020 according to the LAN protocol. The LAN hub 10 creates LAN packets (encapsulating the USB packets 8020) and sends the LAN packets to the attachment unit 110. The attachment unit 110 receives the LAN packets and forwards USB packets 8020 to the LAN computer 130. Referring to FIG. 11G, when the LAN computer 130 receives USB packets 8020 from the LAN hub 10, the USB host software reconstructs an IP packet 8030 from the USB packets 8020.

Detailed Description Text (100):
Referring to FIG. 11B, whenever the LAN computer 130 sends a USB reset signal to the attachment unit 110, the attachment unit 110 sends the reset LAN packet 742 to the LAN hub 10. If the LAN hub 10 receives a corrupted LAN packet (including a corrupted reset LAN packet 742) from the attachment unit, the LAN hub 10 sends the retry LAN packet 740 to the attachment unit 110. Once the LAN hub 10 receives the reset LAN packet 742 without errors, the LAN hub 10 sends the reset LAN packet 742 back to the attachment unit 110. Until the attachment unit 110 receives the reset LAN packet 742 from the LAN hub 10, the attachment unit 110 periodically sends the reset LAN packet 742. Furthermore, until the attachment unit 110 receives the reset LAN packet 742 from the LAN hub 10, the attachment unit 110 only replies to USB packets from the LAN computer 130 with Stall packets. Once the attachment unit 110 is reset, the attachment unit 110 will only respond to USB packets from the LAN computer 130 with a USB device address 0 and control endpoint 0.

Detailed Description Text (101):
Referring to FIG. 11C, if there is a system error (i.e. the LAN hub 10 is in a stall condition, e.g. the LAN hub 10 is not functioning properly) the LAN hub 10 will send a Stall LAN packet 774 to the attachment unit 110 in response to any LAN packet sent by the attachment unit 110. Once the attachment unit 110 receives a stall LAN packet 774 from the LAN hub 10, the attachment unit 110 will send a stall packet to the LAN computer 130 in response to any USB packet from the LAN computer 130. The USB host software in the LAN computer 130 typically informs the client software of the stall condition.

Detailed Description Text (112):
As mentioned earlier, when the LAN computer 130 wishes to communicate with another LAN computer 130 or with a LAN computer 190, 215, 260 or a network device 40 (such as a remote computer), the client software in the LAN computer 130 typically generates IP packets according to the IP protocol. (Other protocols such as the higher layers of Ethernet can be used). (Similarly, when the LAN computer wishes to interact with a USB device 100 or 180, the client software in the LAN computer 130

typically generates IP packets according to the IP protocol). Referring to FIG. 11F, since each IP packet is typically greater than 64 bytes, the USB host software fragments the IP packet 8000 into a plurality of USB packets 8010.

Detailed Description Text (118):
The LAN computer 260 interacts with the communications manager virtual device (CMD) 1020 using this client software (and the host software). The client software communicates with the communicators manager virtual device (CMD) 1020 with IP packets according to the Internet (IP) protocol. As discussed in more detail later, since each IP packet is typically larger than each USB packet, the host software fragments each IP packets into a plurality of USB packets which are sent to the CMD 1020 using the USB protocol. At the CMD 1020, each IP packet is reconstructed from the USB packets. Similarly, the CMD 1020 fragments each IP packet destined to the LAN computer 260 into a plurality of USB packets. The client software reconstructs each IP packet from the USB packets.

Detailed Description Text (119):
The LAN computer 260 interacts with the communications manager virtual device (CMD) 1020 using the client software and the host software to determine what USB devices 100, 180 are available on the LAN hub 10 to "virtually" connect to the LAN computer 260. The client software sends a device directory command to the communications manager virtual device (CMD) 1020. In particular, the device directory command is intercepted by the host software. The host software sends the device directory command to the communications manager virtual device 1020. The communications manager virtual device 1020 forwards the device directory command to the LAN hub 10. In response to the device directory command, the LAN hub 10 sends to the communications manager virtual device 1020 a device listing of all the available USB devices 100 and 180 and their USB device addresses and end points that are connected to the end hubs 80 and composite end hubs 160 (e.g. FIG. 20). The communications manager virtual device 1020 forwards the device listing to the LAN computer 260 over multiple USB packets. A user of the LAN computer 260 selects a USB device 100 or 180 from the listing and the client software informs the USB host software. The USB host software sends a command to the communications device 1020 indicating the USB device 100 or 180 to be "virtually" connected to the LAN computer 260. The enhanced attachment unit 240 informs the LAN hub 10 of the USB device 100 or 180 to be virtually connected to the enhanced attachment unit 240. If the USB device 100 or 180 is still available, the LAN hub 10 informs the enhanced attachment unit 240 that the USB device 100 or 180 has been attached. The LAN hub 10 also informs the enhanced attachment unit 240 whether the USB device is a low speed USB device or a high speed device. Upon regular polling of the enhanced attachment by the LAN computer 260, the enhanced attachment unit 240 will respond with a status change to a previously disconnected USB port 1010 on the enhanced attachment unit 260 (or virtual hub device) (i.e. a USB device is now attached to one of the virtual USB ports 1010). The LAN computer 260 will then send a reset command to the USB device 100, 180 by sending a USB port reset command to the enhanced attachment unit 240 (or the virtual hub) using USB device address 0. The enhanced attachment unit forwards the reset command to the LAN hub 10. The LAN hub 10 forwards the reset command to the USB device 100, 180. Once the USB device 100, 180 has been reset, the LAN computer 260 will send a set-up packet and a data packet containing a first unique USB device address for the USB device 100, 180 to place the USB device 100, 180 in the addressed state. The set-up packet and the data packet are forwarded to the LAN hub 10 using the variant of the USB protocol. It is important to note that the LAN hub 10 typically assigns a second unique (non-zero) USB device address to the USB device 100, 180. (The second USB device address may be different than the first USB address since the first USB device address assigned by the LAN computer 260 may have already been assigned by the LAN hub 10 to another USB device 100 or 180). The LAN computer 260 sends a configuration command for an end point 0 of the USB device 100, 180 to be configured. The enhanced attachment unit 240 forwards the configuration command to the LAN hub 10 using the second USB device address. The LAN hub 10 forwards the configuration command to the USB device 100, 180 using the second USB device address. The LAN hub 10 will also issue setup commands to the enhanced attachment unit 240 to emulate the end point characteristics for that chosen configuration. Referring in particular to FIG. 15I, the LAN hub 10 sends a set-up LAN packet 2500 to the enhanced attachment unit 240. The set-up LAN packet 2500 has a plurality of fields. A first field 2510 indicates the type of packet--a set-up

packet in this case. A second field 2520 contains a set-up token which contains the USB device address of the USB device 100, 180 and the endpoint number of the endpoint being configured. A third field 2530 indicates the maximum length of data that can be transferred to or from the endpoint of the USB device. A fourth field 2540 indicates the type of endpoint--In or Out. A fifth field 2550 indicates whether the endpoint is isochronous or asynchronous. A sixth field 2560 holds the frame number of a future packet on which the specified endpoint will become operational. The set-up packet 2500 also has a CRC 2570 for error checking.

Detailed Description Text (121):
Once the USB device has been configured, any USB packets sent by the LAN computer 260 to the first USB device address will be forwarded to the LAN hub 10 via the enhanced attachment unit 240. The LAN hub 10 forwards the USB packets to the USB device 100, 180 using the second USB device address. Any response from the USB device will be forwarded to the enhanced attachment unit via the LAN hub 10 using the first USB device address. It should be noted that for isochronous transactions, the LAN hub 10 knows the precise schedule of the isochronous transactions, and thus the LAN hub 10 can have data ready for immediate response to an In LAN packet issued by the enhanced attachment unit. For bulk/control/interrupt transactions, the first LAN computer 260 issued IN token packet will by met with a NAK handshake packet; however, the In token packet will be forwarded to the USB device via the LAN hub 10 (using an In LAN packet) and any returned data will be stored in the enhanced attachment unit 240 when the next In token packet is sent (or retried) by the LAN computer 260. Optionally, the LAN hub 10 could poll the appropriate device end points of the USB device with In LAN packets periodically to have data ready for any In LAN packets issued by the enhanced attachment unit. This approach would minimize the number of NAK handshake packets that the LAN computer 260 would encounter in response to In token packets issued by the LAN computer 260.

Detailed Description Text (122):
Once the USB device has been configured, the enhanced attachment unit 240 works very much in a similar way as the attachment unit 110. Data from the client software in the LAN computer 260 intended for a USB device is intercepted by the USB host software in the LAN computer 260. The USB host software creates USB packets containing the data according to the USB protocol. Similarly, USB packets containing data from a USB device are received by the LAN computer 260. The USB host software in the LAN computer 260 extracts data and sends the data to the client software.

Detailed Description Text (126):
Since the client software in the LAN computer 260 generates IP packets according to the IP protocol, the LAN computer 260 can easily communicate with another LAN computer 260 or with a LAN computer 130, 190, 215 or network device 40 (such as a remote computer). (Other protocols, such as the higher layers of Ethernet, can be used). Referring to FIG. 11F, since each IP packet is typically greater than 64 bytes, the USB host software fragments the IP packet 8000 into a plurality of USB packets 8010. The USB packets 8010 are sent via the enhanced attachment unit 240. Similarly, referring to FIG. 11G, when the LAN computer 260 receives USB packets 8020 sent from another LAN computer 260, a LAN computer 130, 215 or 190 or a network device 40 (such as remote computer), the USB host software reconstructs the IP packet 8030 from the USB packets 8020. To allow communication between the LAN computer 260 and a LAN computer 130, 190 or 215 or another LAN computer 260 or a network device 40, the LAN hub 10 will present to the communications manager virtual device (CMD) 1020 (in response to a device directory command) the ability to attach a "virtual modem" device which will work identically as the attachment unit 110. Alternatively, the CMD 1020 will perform the function of a virtual modem since all the packets between the CMD 1020 and the LAN computer 260 are IP packets as previously described.

Detailed Description Text (128):
Referring to FIG. 15B, whenever the LAN computer 260 sends a USB reset command to the enhanced attachment unit 240 or to a USB device 100 or 180 (on a virtual USB port), the enhanced attachment unit 240 sends a reset LAN packet 9200 to the LAN hub 10 using the preferred variant of the USB protocol. The reset LAN packet typically consists of a reset ID 9210 and a field 9220 indicating the port number to which the USB device 100 or 180 is virtually connected. If the port number is 0, an overall

reset for the LAN link 250 and all the virtually connected USB devices occurs. If the LAN hub 10 receives a corrupted reset LAN packet 9200, the LAN hub 10 sends the retry LAN packet 740 to the enhanced attachment unit 240. Once the LAN hub 10 receives the reset LAN packet 9200 without errors, the LAN hub 10 sends the reset LAN packet 9200 to the respective outer hub device. The respective outer hub device then resets the respective USB device. In addition, once the LAN hub 10 receives the reset LAN packet 9200 without errors, the LAN hub 10 sends the reset LAN packet 9200 back to the enhanced attachment unit 240. Until the enhanced attachment unit 240 receives the reset LAN packet 9200 from the LAN hub 10, the enhanced attachment unit 240 periodically sends the reset LAN packet 9200. Furthermore, until the enhanced attachment unit 240 receives the reset LAN packet 9200 from the LAN hub 10, the enhanced attachment unit 240 only replies to USB packets from the LAN computer 260 addressed to the enhanced attachment unit 240 or to the USB device 100 or 180 on a virtual USB port (depending on what was reset) with Stall packets. Once the enhanced attachment unit 240 or the USB device 100, 180 is reset, the enhanced attachment unit 240 or the USB device 100, 180 respectively will respond to USB packets from the LAN computer 260 with the USB device address 0.

Detailed Description Text (129):
Referring to FIG. 15C, if there is a system error (e.g. the LAN hub 10 is not functioning properly) the LAN hub 10 will send a stall LAN packet 774 to the enhanced attachment unit 240 in response to any LAN packet sent by the enhanced attachment unit 240. Once the enhanced attachment unit 240 receives a stall LAN packet 774 from the LAN hub 10, the enhanced attachment unit 240 will send a stall packet to the LAN computer 260 in response to any USB packet from the LAN computer 260 addressed to the enhanced attachment unit 240 or to any of the USB devices 100, 180 that are "virtually" connected to the enhanced attachment unit 240. The USB host software in the LAN computer 260 typically informs the client software of the Stall condition.

Detailed Description Text (150):
Buffers A 5270 typically comprise a temporary buffer A0, a transmit buffer A2, a receive buffer A1, a receive control buffer A1 and a transmit control buffer A2. Similarly, buffers B 5280 typically comprise a temporary buffer B0, a receive buffer B1, a transmit buffer B2, a transmit control buffer B2 and a receive control buffer B1. From the perspective of USB port A 5330, the virtual modem bridge 200 has three end points: control end point 0, end point 1 and end point 2. USB packets are sent to end point 1. USB packets are read from end point 2. To avoid confusion, control end point 0, end point 1 and end point 2 will be called control end point A0, end point A1, and end point A2 respectively. Similarly, from the perspective of USB port B 5340, the virtual modem bridge 200 also has three end points: control end point 0, end point 1 and end point 2. To avoid confusion, control end point 0, end point 1 and end point 2 will be called control end point B0, end point B1, and end point B2 respectively. Each end point A1, A2 has a corresponding buffer--receive buffer A1 and transmit buffer A2 respectively. Similarly, each end point B1, B2 has a corresponding buffer--receive buffer B1 and transmit buffer B2 respectively. Control end point A0 uses two buffers--receive control buffer A1 and transmit control buffer A2. Similarly, control end point B0 uses two buffers receive control buffer B1 and transmit control buffer B2.

CLAIMS:

wherein, for asynchronous communications, the outer hub device examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the first outer hub device examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the second outer hub device examines USB

packets from the other LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within the USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the control logic means examines USB packets from the USB device, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the control logic means examines US Packets from the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the control logic means examines USB packets from the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the control logic means examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein, for asynchronous communications, the control logic means examines USB packets from the first USB host device or the second USB host device, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

wherein the network device communicate with the LAN hub using the LAN protocol or a network protocol; wherein, for asynchronous communications, the outer hub device examines USB packets from the USB device or the LAN computer, generates handshake packets in response to the USB packets according to the USB protocol and ensures that the handshake packets are generated within a USB time limit prescribed by the USB protocol after receiving the USB packets;

(A) Transmitting an Out token packet and a data packet (collectively the "USB packets") from the host computer to a first outer hub device using the USB protocol;

(B) For asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the host computer within a USB time limit prescribed by the USB protocol after receiving the USB packets;

(C) For asynchronous communications, re-transmitting the USB packets from the host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the host computer or if no handshake packet was received by the host computer within the USB time limit prescribed by the USB protocol;

(D) Converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device;

(F) Reconstructing the USB packets from the LAN packet at the second outer hub device;

(G) Transmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the token packet within

the USB time limit prescribed by the USB protocol;

(H) For asynchronous communications, examining the USB packets at the USB device, generating an appropriate handshake packet at the USB device according to the USB protocol and transmitting the handshake packet from the USB device to the second outer hub device within the USB time limit prescribed by the USB protocol after receiving the USB packets;

(I) For asynchronous communications, re-transmitting the USB packets using the USB protocol from the second outer hub device to the USB device ensuring that the data packet follows the token packet within the USB time limit prescribed by the USB protocol if a NAK handshake packet is received by the second outer hub device or if no handshake packet was received by the second outer hub device within the USB time limit prescribed by the USB protocol;

(b) For asynchronous communications, examining the data packet at the second outer hub device, generating an appropriate handshake packet at the second outer hub device according to the USB protocol and transmitting the handshake packet from the second outer hub device to the USB device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the data USB packet; and,

(A) Transmitting an Out token packet and a data packet (collectively the "USB packets") from the host computer to a first outer hub device using the USB protocol;

(B) For asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the host computer using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB packets;

(C) For asynchronous communications, re-transmitting the USB packets from the host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the host computer or if no handshake packet was received by the host computer within the USB time limit prescribed by the USB protocol;

(D) Converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device;

(G) Reconstructing the USB packets from the LAN packet at the second outer hub device;

(H) Transmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the Out token packet within the USB time limit prescribed by the USB protocol;

(I) For asynchronous communications, examining the USB packets at the USB device, generating an appropriate handshake packet at the USB device according to the USB protocol and transmitting the handshake packet from the USB device to the second outer hub device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the USB packets; and,

(J) For asynchronous communications, retransmitting the USB packets from the second outer hub device to the USB device using the USB protocol ensuring that the data packet follows the token packet within the USB time limit prescribed by the USB protocol if a NAK handshake packet is received by the second outer hub device or if no handshake packet was received by the second outer hub device within the USB time limit prescribed by the USB protocol;

(G) Transmitting the In token packet from the second outer hub device to the USB device using the USB packet;

(b) For asynchronous communications, examining the data packet at the second outer hub device, generating an appropriate handshake packet at the second outer hub device according to the USB protocol and transmitting the handshake packet from the

second outer hub device to the USB device using the USB protocol within the USB time limit prescribed by the USB protocol after receiving the data USB packet; and,

(A) Transmitting a token packet and a data packet (collectively the "USB packets") from the first host computer to a first outer hub device using the USB protocol;

(B) For asynchronous communications, examining the USB packets at the first outer hub device, generating an appropriate handshake packet at the first outer hub device according to the USB protocol and transmitting the handshake packet from the first outer hub device to the first host computer using the USB protocol within a USB time limit prescribed by the USB protocol after receiving the USB packets;

(C) For asynchronous communications, re-transmitting the USB packets from the first host computer to the first outer hub device using the USB protocol if a NAK handshake packet is received by the first host computer or if no handshake packet was received by the first host computer within the USB time limit prescribed by the USB protocol;

(D) Converting the USB packets into at least one LAN packet according to a LAN protocol at the first outer hub device;

(G) Reconstructing the USB packets from the LAN packet at the second outer hub device;

(A) Transmitting a token packet and a data packet (collectively the "USB packets") from the first USB host device to a virtual modem bridge using the USB protocol;

(B) For asynchronous communications, examining the USB packets at the virtual modem bridge, generating an appropriate handshake packet at the virtual modem bridge according to the USB protocol and transmitting the handshake packet from the virtual modem bridge to the first USB host device using the USB protocol within a USB time limit prescribed by the USB protocol after receiving the USB packets;

(C) For asynchronous communications, re-transmitting the USB packets from the first USB host computer to the virtual modem bridge using the USB protocol if a NAK handshake packet is received by the first USB host device or if no handshake packet was received by the first USB host device within the USB time limit prescribed by the USB protocol;